CF-004

ChilmAI: A New Daycare Matching System

Zhaohong Sun[1,2] Naoyuki Yamada[2] Yoshihiro Takenami[2] Daisuke Moriwaki[2]

Received: xx xx, xxxx, Accepted: xx xx, xxxx

Abstract: In this paper, we present ChilmAI, our newly developed system designed to address a practical two-sided matching problem: allocating children to daycare centers—a challenge with significant social impact. In collaboration with several municipalities in Japan, our goal is to develop a reliable and trustworthy clearing algorithm tailored to this problem. We propose a novel algorithm that minimizes the number of unmatched children while ensuring stability. Evaluated on real-world datasets, our algorithm outperforms leading commercial software in both the number of children matched and the number of blocking coalitions (measuring the degree of stability). Our findings have been shared with local governments, and Koriyama City has adopted our algorithm in place of existing solutions since November 2024. Furthermore, our model and algorithm have broader applicability to other critical matching markets, including hospital-doctor matching with couples and school choice involving siblings.

Keywords: Stable Matching, Daycare, Siblings, Stability, Fairness

1. Introduction

With the prevalence of dual-career households in recent years, the demand for daycare facilities in Japan, especially in metropolitan areas, has soared. Unfortunately, scarce space and insufficient teachers lead to a long waiting list each year, leaving numerous children unable to enroll in daycare centers. The waiting child problem becomes one of the critical social challenges nowadays (see the press conference by the Japanese Prime Minister on March 17, 2023, available at https://japan.kantei.go.jp/101_kishida/statement/202303/_00015.html).

The daycare matching process has faced criticism due to long waiting lists and the Japanese government has made considerable efforts to address this issue, including improving working conditions and increasing salaries for childcare workers to engage more people in early childhood education, and providing financial assistance for families in need to enlarge their options of affordable daycare centers. Although the number of children on the waiting list significantly decreased recently, the shortage of daycare facilities continues. This is because not all unmatched children are counted in the waiting list, such as those who live near daycare centers with vacant slots but are only willing to attend certain oversubscribed daycare centers and those whose parents have to suspend their jobs or extend their childcare leave. Thus, despite these measures, the waiting child problem remains a major social challenge and long waiting lists have a profound impact on young couples' careers and lives.

The daycare matching problem bears similarities with conventional two-sided matching problems such as college admissions and job hunting, where one side of the market consists of families who submit applications on behalf of their children and the other side consists of daycare centers with limited resources (e.g., room space, teachers). However, the daycare matching market possesses three features that set it apart from classical matching models. These features include i) *transfers* (i.e., some children who are already enrolled prefer to be transferred to other daycare centers), ii) *siblings* (i.e., several children from the same family report joint preferences and only consent to an assignment if all of them are matched), and iii) *transferable quotas* (i.e., a daycare center may partition grades into grade groups and available spots can be used by any child within the same grade group). It is well-known that when couples exist, there may not exist any matching satisfying stability [Roth [1984]] and determining whether there exists a stable matching is NP-complete [Ronn [1990]]. The presence of these complexities poses more significant challenges.

The objective of this research is to develop a trustworthy algorithm to help municipalities tackle the waiting children problem. The key research question is *how to design and implement practical matching algorithms that minimize the number of children on the waiting list in a transparent, stable and computationally efficient manner.*

We next explain why these principles need to be taken into consideration. *Transparency* requires that the public and policymakers can understand and verify how algorithms make decisions. Currently, the daycare matching system that dominates the market does not disclose its code for commercial purposes. Thus, it is unclear why a certain child is assigned to a particular daycare center or why some children are unmatched. *Stability* plays a critical role in the success of many real-world applications, including hospital-residency matching programs and public school choice. In the context of daycare matching, stability can be decomposed into fairness and non-wastefulness, both of which are vital requirements mandated by municipalities. *Computational efficiency* of the matching algorithm is also important, as the number of children participating in the market is large. It usually costs government officers several weeks to calculate and verify the outcomes manually,

[1] Kyushu University, Fukuoka, Japan
[2] CyberAgent, Tokyo, Japan

the process of which is also prone to mistakes. Implementing the algorithm on a computer can yield an outcome in a few seconds or minutes, which would be much more efficient.

In this paper, we first formalize three distinctive features of the market and develop a comprehensive model that also captures important aspects of other matching markets. Next, we propose a practical algorithm based on constraint programming (CP), a powerful technique for solving NP-hard problems. Third, we evaluate the effectiveness of our algorithm using real-world datasets and highlight key insights into factors that can increase the number of matched children. Finally, we release our implementation to support and facilitate future research in related domains.

2. Daycare Matching Procedure in Japan

In this section, we provide a detailed description of the daycare matching procedure. Each municipality operates its own centralized matching system, with specific rules published on its official website. Although matching is conducted monthly, nearly all of daycare slots are allocated at the beginning of April each year. The outcome is determined by a centralized algorithm that takes into account both family preferences over daycares and daycare priorities over children.

Families first submit their preferences to the local municipality on behalf of their children. The matching system collects two types of preference orderings. The first is each child's individual preference ordering, which is a strict ranking over all acceptable daycare centers, where "acceptable" means that the child strictly prefers being matched to any listed daycare over remaining unmatched. The second type is the family-level preference ordering, defined over combinations of daycare centers for all children in the family. If a family's preferences fall into one of several predefined categories, the family only needs to submit their children's individual preferences along with the selected preference type. The system then automatically generates the corresponding family-level preference ordering. For example, one predefined type may require all children to be assigned to the same daycare, while another may prioritize placing the elder sibling first whenever possible. If a family's preferences do not fit any predefined category, they must manually specify a preference ordering over all acceptable daycare combinations for their children.

Each municipality uses its own scoring system to calculate children's priority scores, which determine their relative standing in the matching process. These scores are based on various family circumstances, such as parents' employment status, health conditions, and household composition. For instance, children from low-income households, single-parent families, or families where a guardian is affected by illness or disability are usually given higher priority. To derive a strict priority ordering over all children, municipalities apply tie-breaking rules by adding small fractions to the base scores according to predefined criteria. One common feature we observed is that siblings from the same family typically receive identical scores; however, in certain cases, one sibling may be granted additional points due to specific attribute, e.g., having a disability.

Each daycare derives a strict priority ordering over children based on their priority scores. However, daycares may also assign additional points to certain children under specific conditions. For instance, a child may receive extra points for a particular daycare if they have a sibling currently enrolled there. An important practical rule is that if a child is already enrolled at a daycare, they are given higher priority at that daycare compared to children who are not currently enrolled there.

The Japanese government enforces two types of feasibility regulations: grade-specific minimum room space per child and grade-specific maximum teacher-to-child ratios per teacher. In the current matching system, each daycare sets a hard quota for each grade that complies with these regulations in advance. It is important to note that these quotas are designed to accommodate only new applicants. When a child initially enrolled at a daycare transfers to a different daycare, an additional seat becomes available at the original daycare.

3. Basic Model

In this section, we present a basic model of daycare matching in which each family may have several children.

An instance I^B of the basic daycare matching problem is defined as a tuple $I^B = (C, F, D, Q, \succ_F, \succ_D)$, where C, F, and D denote the sets of children, families, and daycares, respectively; Q is a capacity function assigning each daycare $d \in D$ a capacity Q_d; and $\succ_F$ and $\succ_D$ represent the preference and priority profile of families and daycares, respectively.

Each child $c \in C$ belongs to exactly one family, denoted $f_c \in F$. For each family $f \in F$, let C_f be the set of children belonging to f. We assume that the children in $C_f = \{c_1, c_2, \dots, c_k\}$ are sorted in a fixed order, such as by age. A dummy daycare d_0 is included in D, which represents the option of being unmatched for children. Each daycare $d \in D$ has a strict priority ordering $\succ_d$ over all children C. Each family $f \in F$ has a strict preference ordering $\succ_f$ over tuples of daycare centers, representing its joint preferences for the placements of children in C_f.

Example 1. *Suppose family f has three children, i.e., $C_f = \{c_1, c_2, c_3\}$. A tuple (d_1, d_1, d_2) appearing in $\succ_f$ indicates a configuration where children c_1 and c_2 are assigned to daycare d_1, and child c_3 is assigned to daycare d_2.*

For a child c from family f, we denote by $\succ_c$ the projected preference ordering of child c over daycares, derived from the family's joint preference $\succ_f$. Specifically, if family f has k children, $C_f = \{c_1, \dots, c_k\}$, then the projected preference $\succ_{c_i}$ for child c_i is obtained by taking the i-th entry from each daycare tuple in $\succ_f$. It is important to note that $\succ_c$ does not represent child c's individual preferences, but rather a projection of the family's joint preferences. As a result, the same daycare may appear multiple times in $\succ_c$. We frequently use $\succ_c$ in our algorithmic design and analysis. This is illustrated in Example 2.

Example 2 (Instance of Projected Preferences)**.** *Consider three daycares $D = \{d_0, d_1, d_2\}$ and one family f with two children $C_f = \{c_1, c_2\}$. Family preferences $\succ_f$ and projected preferences of children c_1 and c_2 are as follows:*

$\succ_f$: $(d_1, d_1), (d_2, d_2), (d_1, d_0), (d_2, d_0), (d_0, d_1), (d_0, d_2)$
$\succ_{c_1}$: $d_1, d_2, d_1, d_2, d_0, d_0$
$\succ_{c_2}$: $d_1, d_2, d_0, d_0, d_1, d_2$.

An outcome μ is a matching between children C and daycares D such that: i) Each child $c \in C$ is assigned to one daycare $\mu(c)$ (possibly the dummy daycare d_0); and ii) $\mu(c) = d$ if and only if $c \in \mu(d)$, where $\mu(d)$ is the set of children assigned to d.

An outcome μ is feasible if, for every daycare $d \in D$, the number of assigned children does not exceed its capacity: $|\mu(d)| \leq Q_d$.

Given an outcome μ and a family $f \in F$ with children $C_f = \{c_1, \ldots, c_k\}$, we write $\mu(f) = (\mu(c_1), \ldots, \mu(c_k))$ to denote the tuple of daycare assignments for the children of family f.

4. Properties

In this section, we present several desirable properties that an algorithm should satisfy. These properties have not only been extensively studied in matching theory, but are also practical requirements in real-life matching markets.

One of the most important solution concepts is stability, which refers to a situation in which no pair of agents can jointly improve their outcomes by deviating from the current matching [Gale and Shapley [1962]]. To formalize stability in our setting, we introduce the concept of a choice function for daycares, which provides a convenient way to represent their selection behavior.

Definition 1 (Choice Function Ch_d)**.** *Given a subset of children $C' \subseteq C$, the choice function Ch_d of daycare d selects a subset of C' based on its priority ordering $\succ_d$, choosing children one at a time in order of priority, subject to its capacity constraint Q_d.*

We generalize the notion of stability in Definition 2, characterizing it as the absence of blocking coalitions. Informally, a matching μ is stable if there is no family f and a tuple of daycares $\succ_{f,j}$ (the j-th tuple in $\succ_f$) that would constitute a blocking coalition. Such a coalition arises when: 1) f strictly prefers $\succ_{f,j}$ to its current assignment $\mu(f)$; and 2) each daycare in $\succ_{f,j}$ can accommodate the children in C_f and all other children with higher priority than at least one child in C_f, without violating capacity constraints.

This definition captures the essence of stability and aligns with established notions in related models, such as matching with couples [Ashlagi et al. [2014]], and is closely related to the concept of stability in the matching with contracts framework introduced by Hatfield and Milgrom [2005], where no additional contract is mutually preferable.

Definition 2 (Stability)**.** *Given a feasible outcome μ, family f with children $C_f = \{c_1, \cdots, c_k\}$ and a tuple of daycares $\succ_{f,j} = (d_1^*, \ldots, d_k^*)$ located at position j of $\succ_f$, will form a blocking coalition if i) family f prefers $\succ_{f,j}$ over $\mu(f)$, and ii) $\forall d \in D(f, j)$, $C(f, j, d) \subseteq Ch_d(\mu(d) \cup C(f, j, d))$ where $D(f, j)$ denotes the set of distinct daycares located at position j of $\succ_f$ and $C(f, j, d)$ denotes a subset of children who apply to daycare d w.r.t. $\succ_{f,j}$. A feasible matching is stable if there is no blocking coalition.*

Example 3 (Illustration of Stability)**.** *Consider two families f_1 with children $C(f_1) = \{c_1, c_2\}$ and f_2 with children $C(f_2) = \{c_3\}$. There are two daycares $D = \{d_1, d_2\}$ with one slot each. The preferences and priorities are as follows:*

$\succ_{f_1}$: (d_1, d_2)　$\succ_{d_1}$: c_1, c_3
$\succ_{f_2}$: d_1, d_2　$\succ_{d_2}$: c_3, c_2.

Consider matching μ_1 where family f_1 is matched to (d_1, d_2) while f_2 is unmatched, i.e., $\mu_1(c_1) = d_1$, $\mu_1(c_2) = d_2$, $\mu_1(c_3) = \emptyset$. Matching μ_1 is not stable, because family f_2 can form a blocking coalition with daycare d_2.

More specifically, i) family f_2 prefers d_2 over $\emptyset$ and ii) the notion $C(f_2, 2, d_2) = \{c_3\}$ (i.e., the set of children from family f_2 who applies to daycare d_2 w.r.t. the second element in $\succ_{f_2}$), and $\{c_3\} = Ch_{d_2}(\mu_1(d_2) \cup \{c_3\})$.

Depending on whether certain children are excluded from $\bigcup_{d \in D(f,j)} \mu(d)$, blocking coalitions in Definition 2 can be classified into two categories: justified envy (when some children are removed to create availability) or waste (when there is still enough capacity). Formally,

Definition 3 (Decomposition of Blocking Coalition)**.** *Given a feasible outcome μ, family f with children $C_f = \{c_1, \cdots, c_k\}$ and a tuple of daycares $\succ_{f,j} = (d_1^*, \ldots, d_k^*)$ located at position j of $\succ_f$ form a blocking coalition.*

Let $Re(f, j) = \bigcup_{d \in D(f,j)} \mu(d) \setminus Ch_d(\mu(d) \cup C(f, j, d))$ denote a set of children who are matched to $d \in D(f, j)$ but are not chosen by $Ch_d(\mu(d) \cup C(f, j, d))$, i.e., a set of children who are removed to make room for children C_f.

- *If $Re(f, j) \neq \emptyset$, then family f has justified envy toward $Re(f, j)$.*
- *If $Re(f, j) = \emptyset$, then family f claims that outcome μ is wasteful.*

A feasible matching is fair if it is free of justified envy. A feasible matching is non-wasteful if no family f claims the matching is wasteful.

Fairness is mandated in the daycare matching process under Japanese laws and regulations. In the context of daycare matching (or more generally in two-sided matching markets), fairness is commonly defined as elimination of justified envy. That is, one child with a higher priority score has justified envy toward another child with a lower priority score if the latter is matched to a more preferred daycare center. This fairness concept is widely employed across the country.

Non-wastefulness is advocated in the government plan, suggesting that in addressing the waiting child problem, it's important to make full use of all available resources, which could help to alleviate the shortage of daycare spaces. It states that no family cannot be matched to a more preferred tuple of daycare centers without changing the assignment of any other family.

Both concepts coincide with their counterparts in the classical school choice without siblings and initial enrollments (e.g., Abdulkadiroğlu and Sönmez [2003]).

Example 4 (Decomposition of Blocking Coalition)**.** *Consider the instance in Example 3. Matching μ_1 does not satisfy fairness, because child c_3 has justified envy toward child c_2. Consider matching μ_2 where family f_1 is unmatched while f_2 is matched to d_2, i.e., $\mu_1(c_1) = \emptyset$, $\mu_1(c_2) = \emptyset$, $\mu_1(c_3) = d_2$. Matching μ_2 does not satisfy non-wastefulness, because family f_2 be matched to daycare d_1 without affecting any other family. More specifically, f_2 could form a blocking coalition with daycare d_1 and $Re(f_2, 1) = \emptyset$.*

5. Algorithm Design

In this section, we present our new algorithm that aims to address the daycare matching problem with three particular features. Given the computational intractability results in existing literature [McDermid and Manlove [2010], Manlove et al. [2017]], we utilize constraint programming (CP), a powerful technique for solving NP-hard problems, to develop an efficient and practical solution. Another important reason that we choose the CP solution is to reduce the cost of social implementation. We can represent the problem as a mixed-integer linear program (MILP) and invoke successful solvers for MILPs, which are commercial and expensive. On the other hand, Google OR-Tools is a free and open-source software suite and provides a powerful CP-SAT solver.

We next introduce a set of variables that are used in the CP model. These variables are carefully defined and utilized to facilitate the optimization process.

For each family $f \in F$ and each position $p \in [1, |\succ_f|]$, let $x[f, p]$ denote a binary variable indicating whether family f is matched to the p-th assignment in preference ordering $\succ_f$.

$$x[f, p] = \begin{cases} 1 & \text{if } f \text{ is matched to the } p\text{-th assignment in } \succ_f \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For each child c from family f and each position p, let $x[c, p]$ denote a binary variable that equals $x[f, p]$, indicating whether child c is matched to the p-th element in child c's projected preference $\succ_c$. Note that we use $x[c, p]$ for illustration purposes only, and do not necessarily create these variables in our algorithm.

$$x[c, p] = x[f, p] \tag{2}$$

For each child c from family f and each daycare d that appears in projected preference ordering $\succ_c$, create an integer variable $x[c, d]$ indicating whether child c is matched to daycare d, where $P(c, d)$ denotes a set of positions corresponding to daycare d in $\succ_c$. Using variable $x[c, d]$ is more convenient for calculating the number of children matched to a particular daycare d.

$$x[c, d] = \sum_{p \in P(c,d)} x[c, p] \tag{3}$$

For each family f and each position p in $\succ_f$, create an integer variable $\alpha[f, p]$. This variable indicates whether family f is matched to the i-th assignment in $\succ_f$, where i is no larger than p. In other words, it indicates whether family f is matched to a weakly better assignment than the p-th assignment.

$$\alpha[f, p] = \sum_{i=1}^{p} x[f, i] \tag{4}$$

Before introducing additional variables, we first define several notations. Recall that $\succ_{f,p}$ denotes the p-th tuple in the family f's preference list $\succ_f$, and let $D(f, p)$ be the set of distinct daycares appearing in $\succ_{f,p}$. For each daycare $d \in D(f, p)$, define $C(f, p, d)$ as the set of children from family f who apply for daycare d in $\succ_{f,p}$. Let $C^+(f, p, d)$ denote the set of children who are not from family f and have higher priority than at least one child in $C(f, p, d)$ according to the daycare's priority ordering $\succ_d$.

We create a binary variable $\gamma[f, p, d]$ for each family f, each position p in $\succ_f$ and each daycare $d \in D(f, p)$. This variable indicates whether daycare d can accommodate children $C(f, p, d)$ along with $C^+(f, p, d)$ without exceeding quota Q_d.

$$\gamma[f, p, d] = \begin{cases} 1 & \text{if } \sum_{c \in C^+(f,p,d)} x[c, d] \\ & \quad +|C(f, p, d)| \le Q_d \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

For each family f and each position p in $\succ_f$, create a binary variable $\gamma[f, p]$ indicating whether daycares $D(f, p)$ will choose children C_f simultaneously. That is, $\gamma[f, p]$ equals 1 if and only if $\gamma[f, p, d]$ equals 1 for each distinct $d \in D(f, p)$.

$$\gamma[f, p] = \begin{cases} 1 & \text{if } \forall d \in D(f, p),\ \gamma[f, p, d] = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For each family f and each position p, create a binary variable $\beta[f, p]$ indicating whether family f forms a blocking coalition with daycares $D(f, p)$. By the definition of stability, if family f is matched to some assignment worse than the p-th assignment in $\succ_f$ (i.e., $\alpha[f, p] = 0$) and daycares $D(f, p)$ can accommodate C_f along with children with higher priority (i.e., $\gamma[f, p] = 1$), then family f and daycares $D(f, p)$ form a blocking coalition.

$$\beta[f, p] = \begin{cases} 1 & \text{if } \alpha[f, p] = 0 \text{ AND } \gamma[f, p] = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We enforce feasibility with the following two constraints. The first constraint, labeled as Constraint 8, specifies that each family can be matched with at most one assignment. The second constraint, labeled as Constraint 9, mandates that for each daycare d, the number of assigned children must not exceed the transferable quota Q_d.

$$\sum_{p=1}^{|\succ_f|} x_{f,p} \le 1 \qquad \forall f \in F \tag{8}$$

$$\sum_{c \in C} x_{c,d} \le Q_d \qquad \forall d \in D \tag{9}$$

As a stable outcome may not always be theoretically guaranteed, our CP algorithm is specifically designed to prioritize two main objectives. First, it aims to identify the minimum number of blocking coalitions, and subsequently, find a matching that maximize the number of matched children with the minimum number of blocking coalitions. This idea is inspired by Manlove et al. [2017]. We design two objective functions in our algorithm. The primary objective in Formula 10 is to discover a feasible matching with the least number of blocking coalitions. Let θ denote the minimum number of blocking coalitions among all such matchings. The secondary objective in Formula 11 is to determine a feasible matching that incorporates the maximum number of matched children, subject to the condition that the number of blocking coalitions is no more than θ. Note that we need to exclude the number of children who are matched to the dummy daycare d_0.

$$\min \sum_{f \in F} \sum_{p \in [|\succ_f|]} \beta[f, p] \tag{10}$$

$$\max \sum_{c \in C} \sum_{p=1}^{|\succ_{f,c}|} x_{c,p} - \sum_{c \in C} \sum_{p' \in P(c,d_0)} x_{c,p'} \tag{11}$$

	Tama-21		Tama-22		Shibuya-21		Shibuya-22		Koriyama-22	
	current	CP	current	CP	current	CP	current	CP	current	CP
# matched children	558	560	464	470	1307	1307	1087	1087	979	1200
# blocking coalition	26	0	2	0	0	0	0	0	1059	0

Table 1: Comparison with Status Quo Methods

	Tama-21		Tama-22		Shibuya-21		Shibuya-22		Koriyama-22	
	CP	CP-Ind	CP	CP-Ind	CP	CP-Ind	CP	CP-Ind	CP	CP-Ind
# matched children	560	571	470	478	1307	1335	1087	1135	1200	1217
# running time	9.6s	12.5s	1.7s	3.0s	18.1s	1000s	11.6s	803s	18.5s	24.1s

Table 2: Impact of Indifferences. CP-Ind denotes the variant with indifferences allowed. CP-Ind found an optimal solution for all datasets within 1000s except for Shibuya-21.

6. Empirical Evidence

In this section, we evaluate the performance of our new algorithm through experiments on several real-life data sets. Our experiment comprises two parts: Firstly, we compare our algorithm with currently deployed algorithms in terms of two principal objectives of the daycare matching problem: the number of matched children and the number of blocking coalitions. Experimental results show that our algorithm consistently outperforms the existing methods, especially a commercial software that dominates the current daycare matching market in Japan. Secondly, we examine the effect of different factors on the number of matched children, such as allowing indifferences in daycares' priorities and permitting a small number of blocking coalitions.

Methodology We implemented our CP model using the Google OR-Tools API. All the experiments were performed on a laptop equipped with an M1-max CPU and 32GB of RAM. We wrote our code in Python and adapted it for the CP-SAT solver offered in the OR-Tools package, which is a powerful and award-winning solver for constraint programming problems. Please refer to the international constraint programming competition (MiniZinc Challenge) for more details.

The data sets we tested are provided by three municipalities: Shibuya, a major commercial and finance center in Tokyo; Tama, a suburban city located in the west of Tokyo; and Koriyama, a large city located in the northern region of Japan. While our CP algorithm is capable of handling transferable quotas, it is important to note that we only have access to a relatively small data set that allows for transferable quotas. Unless explicitly stated otherwise, the following experimental results will focus on data sets that include only two features: siblings and initial enrollments. The number of children participating in these data sets varies from 550 to 1589 and the number of daycares varies from 33 to 86.

6.1 Experiment 1: Comparison with Status Quo Methods

For Shibuya and Koriyama data sets, the status quo algorithm is provided by a commercial and non-disclose software. However, Tama also acquired the same software but ultimately disregarded it due to its unsatisfactory performance, and currently employs a manual approach instead. In Table 1, we summarize the results of experiments comparing these methods. Notably, the CP algorithm not only increases the number of matched children by up to 23%, but also generates outcomes with the minimum number of blocking coalitions, ensuring a more satisfactory outcome with respect to fairness and non-wastefulness. In the next experiment, we further show that it is still possible to decrease the number of unmatched children while maintaining stability.

6.2 Experimental 2: Effect of Different Factors on the Outcomes

Next, we present a summary of the findings regarding the impact of different factors on the number of matched children. 1) **Indifferences** In the current daycare matching process, each municipality employs a distinct and intricate scoring system to establish the priority order for children. However, tie-breakers are frequently employed by introducing a small fractional value to derive a strict priority ordering. An alternative approach we propose is to remove the small fractional values and allow for indifferences in daycare priorities. Our research summarized in Table 2 reveals that incorporating indifferences can lead to a substantial decrease in the number of unmatched children, while still maintaining stability. It is important to note that computing such a stable matching with maximum matches may require considerably more time in certain cases. 2) **Blocking Coalition** Throughout our tests on all the data sets, we consistently achieved stable outcomes, wherein no blocking coalitions were present. However, we wanted to explore the possibility of allowing a small number of blocking coalitions to potentially benefit more children. Surprisingly, our investigations revealed that incorporating a small number of blocking coalitions, such as 5, did not necessarily lead to a significant increase in the number of matched children. In contrast, a larger number of blocking coalitions, such as 100, showed a noticeable impact on increasing the number of matched children. However, it is important to note that as the number of permitted blocking coalitions increases, the computational time required to find a feasible solution also increases significantly. 3) **Transferable Quotas** We only have one data set in which transferable quotas are allowed. Although the use of transferable quotas could lead to an increase of matched children in theory, we cannot draw a clear conclusion on the effect of transferable quotas due to the small size of the data set.

7. Extension of The Basic Model

In this section, we extend the basic model to incorporate two additional features of the market. First, some children may be initially enrolled at a daycare $d \in D \setminus \{d_0\}$ and prefer to be transferred to a different one. We use the function $\omega(c) \in D$ to denote the initial daycare of child c. If $\omega(c) = d_0$, then child c is considered a new applicant. We extend the concept of $\omega(c)$ to $\omega(f)$ for a family f with children $C_f = \{c_1, \ldots, c_k\}$ by defining $\omega(f) = (\omega(c_1), \ldots, \omega(c_k))$ as the initial enrollments of family f.

Second, there is a set of grades G and each child c is associated with one grade denoted by $G(c)$. Let Q represent grade-specific quotas at all daycares, and specifically $Q(d, g)$ represents the quota for grade g at daycare d. A daycare center may partition grades G into disjoint grade groups and available spots can be used by any child within the same grade group, which allows for greater flexibility in the matching process. Formally, for each daycare d, the set $G(d) = \{G_{d_1}, \cdots, G_{d_l}\}$ is a partition of grades G into disjoint grade groups. , such that i) the union of all groups is equal to G, which means that all grades are covered by the grade groups, i.e., $\bigcup_{\widetilde{g} \in \widetilde{G}(d)} \widetilde{g} = G$, and ii) any two groups are disjoint, meaning that no two grade groups overlap.

Given a grade group $G_d \in G(d)$ at daycare d, let $Q(d, G_d) = \sum_{g \in G_d} Q(d, g)$ denote a transferable quota, which is the sum of grade-specific quotas of each grade $g \in G_d$. The use of transferable quotas has been proposed in previous work [Kamada and Kojima [2023]] and is also employed by some daycare centers in practice. For example, in Tama City, the majority of daycare slots are allocated at the beginning of April each year, following strict grade-specific quotas. In the following months, only a limited number of slots become available, and a few daycare centers allow for transferable quotas to accommodate these cases.

Example 5 (Transferable Quotas)**.** *Suppose daycare d partitions grades G into* $G(d) = \{\{0\}, \{1, 2\}, \{3, 4, 5\}\}$. *It indicates that grade* 0*, grades* $\{1, 2\}$ *and grades* $\{3, 4, 5\}$ *form three grade groups respectively, where grade-specific quotas are transferable within each grade group. If daycare d does not permit transferable quotas, then each grade itself is considered a grade group, i.e.,* $G(d) = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$.

An outcome, denoted by μ, represents a matching between children C and daycares D such that each child c is matched to one daycare $d \in D$. We use $\mu(c)$ to denote the daycare matched to child c in the outcome μ. Similarly, $\mu(d)$ represents the set of children matched to daycare d in μ. To specify the set of children of a particular grade g who are matched to daycare d in μ, we use the notation $\mu(d, g)$. Furthermore, $\mu(d, G_d)$ refers to the set of children belonging to the grade group G_d who are matched to daycare d in μ. Additionally, $\mu(f)$ is a tuple of daycares matched to children C_f in μ. Note that $C_f = \{c_1, \ldots, c_k\}$ is sorted in a predetermined order, and $\mu(f) = (d_1^*, \ldots, d_k^*)$ indicates that child c_i is matched with daycare d_i^*, respectively.

8. Conclusion

In this paper, we describe three key features of the daycare matching problem in Japan, which consolidate and integrate other important matching models such as hospital-doctor matching with couples. We propose a practical algorithm based on constraint programming (CP) that maximizes the number of children matched to daycare centers while ensuring stability, a desirable property in the context of matching problems under preferences. Through experimentation, we have discovered that our CP algorithm holds great potential in significantly improving the process of matching children to daycare centers when compared to the current methods employed in practice. This algorithm presents a promising approach to addressing the daycare matching problem and has the capacity to shed light on similar matching problems.

References

A. Abdulkadiroğlu and T. Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

I. Ashlagi, M. Braverman, and A. Hassidim. Stability in large matching markets with complementarities. *Operations Research*, 62(4):713–732, 2014.

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

J. W. Hatfield and P. R. Milgrom. Matching with contracts. *American Economic Review*, 95(4):913–935, 2005.

Y. Kamada and F. Kojima. Fair matching under constraints: Theory and applications. *The Review of Economic Studies*, 2023. ISSN 0034-6527.

D. F. Manlove, I. McBride, and J. Trimble. "Almost-stable" matchings in the hospitals/residents problem with couples. *Constraints*, 22(1):50–72, 2017.

E. J. McDermid and D. F. Manlove. Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization*, 19(3):279–303, 2010.

E. Ronn. NP-complete stable matching problems. *Journal of Algorithms*, 11(2):285–304, 1990.

A. E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 92(6):991–1016, 1984.